REMARKS

Claims 1-3, 9, 10, 12-16, 20-22, 26, and 28-30 stand rejected under 35 U.S.C. 103(a) as unpatentable over US Publication 2004/0044997 by Talati (Talati) in view of US Patent 6,085,333 to DeKoning et al. (DeKoning), US Publication 2003/0092438 by Moore et al. (Moore) and US Patent 6,658,659 to Hiller (Hiller). Claims 11, 18, and 23 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Talati in view of DeKoning, Moore, Hiller, and US Patent 6,986,132 to Schwabe (Schwabe). Claims 31-36 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Talati in view of DeKoning, Moore, Hiller, and US Publication 2003/0120909 by Zimmer et al. (Zimmer).

Applicants thank the Examiner for the telephone interview of September 25, 2009. We discussed the present invention and a proposed amendment. The Examiner suggested additional limitations including specifying Microcode Reconstruct and Boot format, that the code is embedded code in a host bus adapter, and details of image version numbers. Applicants further thank the examiner for the additional guidance provided on October 21, 2009. Applicants have amended the claims with the suggested limitations with minor changes.

Amendments to the claims

Applicants have amended claim 1 with the limitations "…An apparatus for updating an embedded code image in a host bus adapter performing high speed data transfer between a host system and a storage system, comprising a processor, a main memory, and a temporary memory, the main memory and the temporary memory storing Microcode Reconstruct and Boot (MRB) format code images, and the apparatus further comprising…." The amendment is well supported by the specification. See page 9, ¶ 33 (host bus adapter); page 10, ¶ 35-36; fig. 2, ref. 202, 204,

208 (processor, main memory, temporary memory); page 16, ¶ 61 (Microcode Reconstruct and Boot format).

Applicants have further amended claim 1 with the limitations "...an old code image comprising a first MRB format header..." and "...a new code image comprising a second MRB format header...." The amendment is well supported by the specification. See page 16, ¶ 61; fig. 4, ref. 410, 416.

Claim 1 is also amended with the limitation "...accessing an old code image version number pointed to by the first MRB format header and a new code image version number pointed to by the second MRB format header...." The amendment is well supported by the specification. See page 16, ¶ 61; page 17, ¶ 63.

Applicants have further amended claim 1 with the limitation "...incompatibilities between the old code image and the new code image from the old code image version number and the new code image version number~~version information~~...." The amendment is well supported by the specification. See page 17, ¶ 64. In addition, claim 1 is amended with the limitation "...a difference in initialization requirements for storage registers, memory, and hardware devices...." The amendment is well supported by the specification. See page 11, ¶ 41.

Independent claims 10, 13, 20, 29, and 30 are similarly amended. Claims 31-36 are amended to conform to amended predecessor claims.

Response to rejections under 35 U.S.C. § 103

Claims 1-3, 9, 10, 12-16, 20-22, 26, and 28-30 stand rejected under 35 U.S.C. 103(a) as unpatentable over Talati in view of DeKoning, Moore, and Hiller. Claims 11, 18, and 23 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Talati in view of DeKoning, Moore,

Hiller, and Schwabe. Claims 31-36 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Talati in view of DeKoning, Moore, Hiller, and Zimmer.

Independent claim 1 as amended includes the limitations:

"1. An apparatus for updating an embedded code image in a host bus adapter performing high speed data transfer between a host system and a storage system, comprising a processor, a main memory, and a temporary memory, the main memory and the temporary memory storing Microcode Reconstruct and Boot (MRB) format code images, and the apparatus further comprising:

the processor executing executable code stored on the main memory occupied by and used by an old code image comprising a first MRB format header, the executable code comprising:

a loader stored in the main memory and loading a new code image comprising a second MRB format header into the temporary memory;

a branch module stored in the main memory causing the processor to **execute a bootstrap module within the new code image**;

the bootstrap module accessing an old code image version number pointed to by the first MRB format header and a new code image version number pointed to by the second MRB format header;

the bootstrap module identifying:

incompatibilities between the old code image and the new code image from the old code image version number and the new code image version number;

a difference in initialization requirements for storage registers, memory, and hardware devices; and

a difference in size and location between the old code image and the new code image;

the bootstrap module further:

accessing capability information for the old code image and capability information for the new code image;

identifying a difference between the capability information; and

reconciling the incompatibilities by

changing an initialization order;

converting a format of a data structure of the old code image to a format compatible with a data structure of the new code image; and

associating persistent data of the old code image with the new code image, such that the persistent data is available in response to execution of a run-time segment of the new code image; and

a copy module copying the new code image into the main memory space occupied by the old code image." Emphasis added.

Independent claims 10, 13, 20, 29, and 30 include similar limitations. The present invention claims a bootstrap module contained in the new code image that identifies incompatibilities and differences between the old and new code images, identifies differences between compatibility information, reconciles the incompatibilities between the old and new

code images, and associates persistent data of the old code image with the new code image. See claim 1.

Applicants submit that claim 1 is distinguished from Talati, DeKoning, Moore, and Hiller by claiming "…execute a bootstrap module within the new code image …," "…the bootstrap module accessing an old code image version number pointed to by the first MRB format header and a new code image version number pointed to by the second MRB format header…," "…the bootstrap module identifying:…a difference in initialization requirements for storage registers, memory, and hardware devices…," "…a difference in size and location between the old code image and the new code image…," and "…reconciling the incompatibilities by… changing an initialization order…."

The bootstrap module of the present invention is embodied in the new code image, with the bootstrap module identifying and reconciling differences while executing from the new code image. Thus the invention need not rely on external code to identify and reconcile differences between old and new code images as taught in Talati and DeKoning, but includes within the bootstrap module of the new code image the functionality to identify and reconcile differences. This for example could allow the new code to be tailored to handle all differences. In contrast, Talati teaches that the current firmware (old code image) is executed until the new firmware version (new code image) is loaded, and then executing the new firmware version. Talati, page 3, ¶ 32. Applicants therefore submit that Talati and also DeKoning, Moore, and Hiller do not teach "…execute a bootstrap module within the new code image …" as claimed in claim 1.

Claim 1 also includes the limitation of the bootstrap module accessing an old code image version number pointed to by the first MRB format header and a new code image version number pointed to by the second MRB format header. As we discussed, this element is not

disclosed by Talati, DeKoning, Moore, and Hiller, which do not disclose the MRB format header.

Claim 1 is amended with the limitation "...the bootstrap module identifying...a difference in initialization requirements for storage registers, memory, and hardware devices...." As we discussed, while DeKoning teaches synching configuration parameters, DeKoning does not teach identifying a difference in initialization requirements for storage registers, memory, and hardware devices. DeKoning, col. 11, lines 24-35. Talati, Moore, and Hiller also do not disclose these elements.

Claim 1 also includes the elements "...a difference in size and location between the old code image and the new code image...." The Examiner points out that Hiller teaches different versions performing same functions in different ways. Office Action of August 12, 2009 (OA), page 7, lines 1-7; citing Hiller, col. 6, lines 12-26. Applicants submit that Hiller does not teach identifying a difference in size and location between old and new code images. Instead Hiller acknowledges that different versions of code have differences, but uses version numbers rather than size and location to identify differences. Hiller, col. 6. Lines 27-30. Applicants therefore submit that Hiller, and also Talati, DeKoning, and Moore do not disclose the element "...a difference in size and location between the old code image and the new code image...."

Claim 1 also claims "...reconciling the incompatibilities by... changing an initialization order...." The Examiner notes that DeKoning teaches reconfiguring a spare controller cache memory so that the cache is similar to a native controller's configuration and then purging the spare controller cache to initialize the cache. OA, page 5, lines 4-8; citing DeKoning col. 10, lines 24-32. However, DeKoning does not change an initialization order to reconcile incompatibilities but instead teaches initializing the spare controller including purging the spare

controller's cache. Applicants therefore submit that DeKoning and also Talati, Moore, and Hiller do not teach the element "...reconciling the incompatibilities by...changing an initialization order...."

Because Talati, DeKoning, Moore, and Hiller do not teach each element of independent claim 1, Applicants submit that claim 1 is allowable. Applicants further submit that claims 10, 13, 20, 29, and 30 are allowable for the same reasons. Furthermore, Applicants submit that dependent claims 2, 3, 9, 11, 12, 14-18, 21-23, 28, and 31-36 are allowable at least due to their dependency from independent claims 1, 10, 13, 20, 29, and 30.

CONCLUSION

Applicants submit that the remarks and amendments put the present application in condition for allowance. In the event the Examiner finds any remaining impediment to the prompt allowance of any of these claims, which could be clarified in a telephone conference, the Examiner is respectfully urged to initiate the same with the undersigned.

Respectfully submitted,

/Brian C. Kunzler/

Brian C. Kunzler
Reg. No. 38,527
Attorney for Applicants

Date: November 12, 2009

Kunzler & McKenzie
8 East Broadway, Suite 600
Salt Lake City, UT 84111
Telephone (801) 994-4646
Fax (801) 531-1929